

Graph based Version of AHC and KNN Algorithm in Combination of Text Segmentation with Text Clustering

Taeho Jo
President
Alpha AI Research
Cheongju, South Korea
tjo018@naver.com

Abstract—In this research, we propose the versions of the KNN algorithm and the AHC algorithm which process graphs for automating the text segmentation by combining itself with the text clustering. In the previous work, even if the KNN version was proposed as an approach to the text segmentation, a domain should be presented as an input text tag. In this research, text subgroups are built based on their contents by the text clustering, and adjacent paragraphs as text partitions are classified into boundary or continuance by nominating a classifier which corresponds to its most similar cluster. The graph based AHC algorithm, and the graph based KNN algorithm are adopted respectively for the text clustering and the text segmentation. The goals of this research are to automate the text segmentation and to improve the performance of both tasks, using the two graph-based versions.

I. INTRODUCTION

The text segmentation is defined as the logical segmentation of a text based on the topic transition between paragraphs. The task is mapped into the classification of an adjacent paragraph pair into boundary or continuance, to judge whether a boundary is put between paragraphs, or not. The process of text segmentation is to partition an input text into paragraphs, generate adjacent paragraph pairs by sliding the two-sized window on the paragraphs, and classify adjacent paragraph pairs into one of the two categories. A boundary is put between paragraphs in each pair which is classified into boundary, and a text is segmented into subtexts following the boundaries. Subtexts as the output of the text segmentation are treated as independent texts.

This research is motivated by defining the text segmentation as a binary classification which corresponds to a domain. It is required to predefine domains manually for designing the text segmentation system. It is composed with binary classification systems as many as domains. The excellent performances of the graph based KNN algorithm and the graph based AHC algorithm was discovered in the text segmentation and the text clustering. In this research, we connect the text segmentation with the text clustering and adopt the graph-based machine learning algorithms for implementing both tasks.

In this research, we propose that the text segmentation should relate to the text clustering, and the graph based

KNN algorithm and the graph based AHC algorithm are applied to both tasks. In this research, we prepare the text collection where each text is segmented into subtexts by the topic transition. The similarity metric between two graphs is defined, and the KNN algorithm and the AHC algorithm are modified into the graph-based versions. The collection is segmented into clusters by the text clustering, and in each cluster, which indicates a domain, we gather paragraph pairs which are labeled with one of the two categories and train its corresponding binary classifier. A domain is automatically decided to an input text by its similarities with the cluster prototypes for selecting a corresponding binary classifier.

Let us mention some benefits from this research. The main problems which are caused by encoding texts into numerical vectors are solved by encoding them into graphs. The performances of the text segmentation and the text clustering are improved by adopting the versions of the KNN algorithm and the AHC algorithm which process directly graphs. It does not require to define domains depending on prior knowledge or subjective views by automating the domain predefinition through the text clustering. It does not require to tag an input text with its domain manually by computing its similarities with the cluster prototypes.

Let us mention some remaining tasks as the further discussions on this research. More advanced machine learning algorithms and deep learning algorithms are modified into graph-based versions. We need to define and characterize mathematically more semantic operations on graphs. The text segmentation system which relates to the text clustering may be implemented as a real program or a library. The text classification and the text clustering relate to the text segmentation for improving their performances.

II. PROPOSED WORKS

A. Encoding Proccerss

This section is concerned with the graph as a text representation and the similarity between graphs. In representing a text into a graph, a word is given as a vertex, and a similarity between words is given as an edge. A graph is viewed as a set of edges, and the similarity between edges is computed before computing the similarity between graphs.

The similarity between them is computed by averaging the similarities of all possible edge pairs. This section is intended to describe the process of encoding a text into a graph and the process of computing the similarity between graphs.

The process of encoding a text into a graph is illustrated in Figure 1. Words are retrieved as the vertices from the text by indexing it. The similarities among words are computed as edges in the edge definition. In the edge selection, the edges with their higher similarities are selected for representing a text into a graph. Refer to [1] for studying the encoding process in more detail.

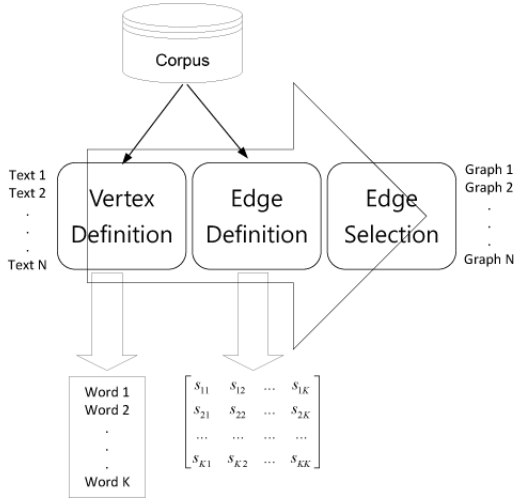


Figure 1. Process of Encoding Texts into Graphs

The three cases of computing the similarity between two edges are illustrated in Figure 2. Each weight which is associated with its own edge is assumed to be a normalized value between zero and one, and if both nodes are identical to each other, the average over two weights is the similarity between two edges. If either of the two nodes is identical to each other, the product of two weights is the similarity between two edges. If neither of two nodes is identical, zero is the similarity between them. The similarity between two edges is always a normalized value between zero and one.

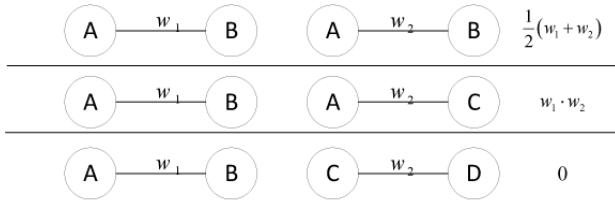


Figure 2. Three Cases of Comparing Two Edges with Each Other

Let us mention the process of computing the similarity between graphs as a semantic similarity between texts. The two graphs which represent texts are notated by edge sets, $G_1 =$

$\{e_{11}, e_{12}, \dots, e_{1|G_1|}\}$ and $G_2 = \{e_{21}, e_{22}, \dots, e_{2|G_2|}\}$. Two edges, $e_{1i} \in G_1$ and $e_{2j} \in G_2$, are associated with their own weights, w_{1i} and w_{2j} , and the similarity between two edges, e_{1i} and e_{2j} by equation (1), (2), or (3), depending on the cases which are mentioned above,

$$\text{sim}(e_{1i}, e_{2j}) = \frac{1}{2}(w_{1i} + w_{2j}) \quad (1)$$

$$\text{sim}(e_{1i}, e_{2j}) = w_{1i} \times w_{2j} \quad (2)$$

$$\text{sim}(e_{1i}, e_{2j}) = 0 \quad (3)$$

The similarity between two graphs is computed by equation (4),

$$\text{sim}(G_1, G_2) = \frac{1}{|G_1| \times |G_2|} \sum_{i=1}^{|G_1|} \sum_{j=1}^{|G_2|} \text{sim}(e_{1i}, e_{2j}). \quad (4)$$

The value which is generated from equation (4), is always given as a normalized value between zero and one as shown in equation (5),

$$0 \leq \text{sim}(G_1, G_2) \leq 1. \quad (5)$$

Let us make some remarks on the encoding process and the computation of the similarity between texts. A text is encoded into a graph by the vertex definition, the edge definition, and the edge weighting. The three cases are considered for computing the similarity between edges. The similarity between graphs is computed by equation (4). In the future research, we consider representing a text into multiple graphs.

B. Graph based KNN Algorithm

This section is concerned with the graph based KNN algorithm. It was initially proposed as the approach to the word classification by Jo in 2018 [2]. The similarity metric between graphs which was described in the previous section is used for computing the similarity between a novice graph and a training example. The labels of its nearest neighbors are voted with their identical weights for deciding the label of a novice input. This section is intended to describe the initial version of KNN algorithm from which its variants are derived.

The process of computing the similarity between a novice item and a training example is illustrated in Figure 3. The labeled words which are gathered as samples are encoded into graphs, and they are notated by a set, $Tr = \{G_1, G_2, \dots, G_N\}$. A novice word is encoded into a graph notated by G . Its similarities with the graphs which represent the sample words are computed by equation (4), as $\text{sim}(G, G_1), \text{sim}(G, G_2), \dots, \text{sim}(G, G_N)$. Some training examples which are most similar as the novice input are selected as its nearest neighbors.

In Figure 4, the process of selecting nearest neighbors by the ranking selection is illustrated. The training examples are sorted by the descending order of their similarities, after

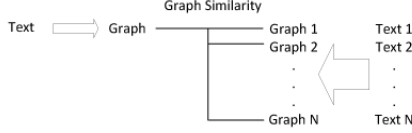


Figure 3. Similarities of Novice Input with Training Examples

computing their similarities with a novice example. The training examples with their higher similarities with a novice example are selected as its nearest neighbors, as expressed in equation (6),

$$Nr = \text{Select}_k(Tr, G), Nr \subseteq Tr \quad (6)$$

The set of nearest neighbors, Nr , is composed with elements as expressed in equation (7),

$$Nr = \{G_1^{near}, G_2^{near}, \dots, G_k^{near}\} \quad (7)$$

The elements in the set, Nr , are used for voting their labels in the next step.

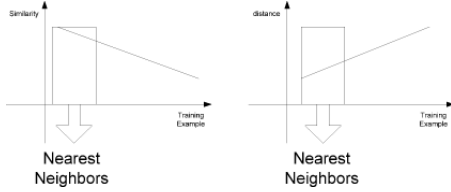


Figure 4. Selection of Most Similar or Least Distant Training Examples as Nearest Neighbors

The process of voting the labels of nearest neighbors for decoding the label of a novice word is illustrated in Figure 5. The predefined categories are notated by $c_1, c_2, \dots, c_m$, and the label of a nearest neighbor is notated by $c_j = \text{label}(G_i^{near})$. The number of nearest neighbors which belong to the category, c_j , is notated by $\text{Count}(Nr, c_j)$. The label of the novice item, G , is decided by equation (8),

$$\text{label}(G) = \arg\max_{j=1}^m \text{Count}(Nr, c_j) \quad (8)$$

The process of deciding the label of a novice item by equation (8), is called voting.

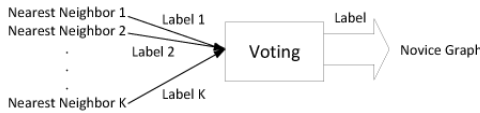


Figure 5. Voting of Labels of Nearest Neighbors for Deciding Label of Novice Input

Let us make some remarks on the graph based KNN algorithm. It computes the similarities of a novice item with the training examples. The training examples with their highest similarities with the novice item are selected as its

nearest neighbors. The label of the novice item is decided by voting the labels of its nearest neighbors. The ranking selection is adopted for selecting the nearest neighbors from training examples, in this version.

C. System Architecture

This section is concerned with the text segmentation system which adopts the graph based KNN algorithm. In Section II-B, we described the proposed version of KNN algorithm as the approach to the text segmentation. It is viewed as a binary classification which classifies a paragraph pair into boundary or continuance. This section is intended to describe the text segmentation system with respect to its function and its architecture.

The process of gathering sample paragraph pairs and classifying a novice one, is illustrated in Figure 6. Because even a same paragraph pair may be classified differently depending on the domain, the task is called domain dependent classification. Sample paragraph pairs are gathered domain by domain, and each pair is labeled with boundary or continuance. The paragraph pairs are taken from texts which are tagged with their same domain, each paragraph pair is classified into boundary or continuance. Sample paragraph pairs which are labeled with one of the two categories are encoded into graphs in each domain.

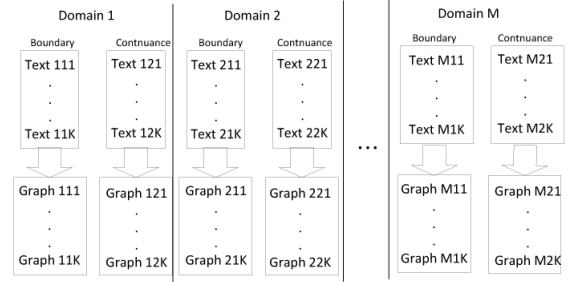


Figure 6. Preparation of Training Examples

The entire architecture of the proposed text segmentation system is illustrated in Figure 7. A text is given as the input, and adjacent paragraph pairs are extracted by partitioning the text into paragraphs and slide the two sized window on them. The sample paragraph pairs in the boundary group and the continuance group and ones which extracted from the text are mapped into graphs in the encoding module. The paragraph pairs from the text are classified into one of the two categories in the similarity computation module and the voting module. A boundary is put between paragraphs in each pair which is classified into boundary in the text, and it is partitioned into subtexts by the boundary.

The execution flow of the text segmentation system is illustrated in Figure 8. Texts with same domain are initially collected, adjacent paragraph pairs are extracted from them, and the paragraph pairs are labeled with boundary or continuance. From an input text, adjacent paragraph pairs are

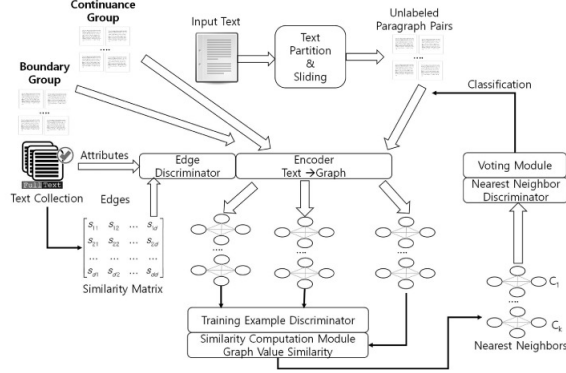


Figure 7. System Architecture

extracted, and are encoded into graphs, together with the sample paragraph pairs. The adjacent paragraph pairs are classified into boundary or continuance, and there are two groups of paragraph pairs: boundary group and continuance group. The boundary is marked between paragraphs in each pair in the boundary group, and text is partitioned by the marked boundaries into subtexts as the final output of this system.

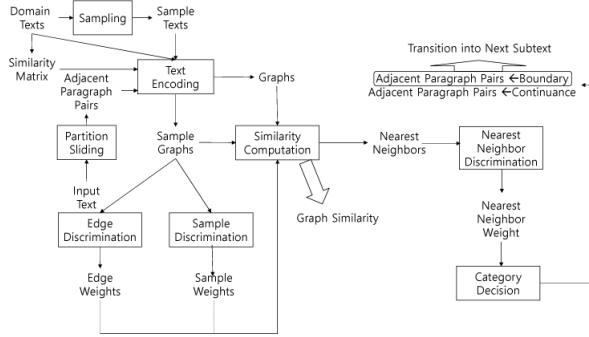


Figure 8. System Flow

Let us make some remarks on the proposed system which is illustrated in Figure 7 as its architecture. The text segmentation is defined as the binary classification where each adjacent paragraph pair is classified into boundary or continuance. The paragraph pairs are encoded into graphs instead of numerical vectors, and they are classified into directly. The input text is partitioned into subtexts by the boundary which is put between paragraphs which are classified into boundary. The subtexts as the semantic division of the input text are treated as independent texts.

D. Connection with Text Clustering

This section is concerned with the connection of the text segmentation with the text clustering. In the previous section, we mentioned the text segmentation as an instance of domain dependent classification. The domains are automatically constructed from the text collection by clustering

texts. The AHC algorithm which is proposed in [4] is used for implementing the text clustering. This section is intended to describe the text segmentation system which is connected with the text clustering for automating the construction of domains.

The architecture of the text clustering system which was proposed in [4] is illustrated in Figure 9. The texts in the group are encoded into graphs, and the AHC algorithm which is proposed in [4] is adopted as the approach. It starts with singletons as many as items, computes the similarities of all possible cluster pairs, and merge the cluster pair with its highest similarity into one cluster. The two steps are iterated until the number of clusters reach the desired number. We may consider replacing the AHC algorithm by its variants for improving the speed and the performance.

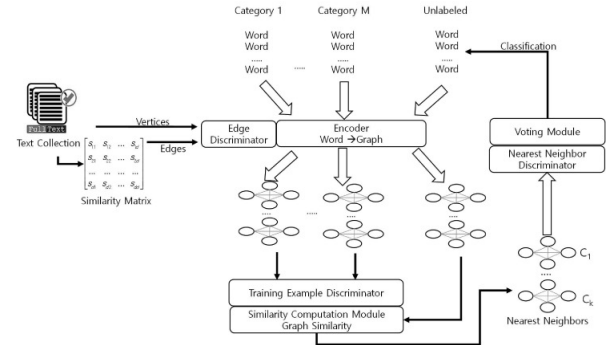


Figure 9. Text Clustering System

The connection of the text segmentation with the text clustering is illustrated in Figure 10. The M domains are defined by clustering texts in a group, initially and automatically. A novice text is given as the input, and its domain, domain D , is decided by its similarities with domains. A classifier which corresponds to domain D is nominated and classify each adjacent paragraph pair in a novice text into boundary or continuance. The M index optimization systems are viewed as independent ones, each of which classifies each paragraph pair into one of the two categories.

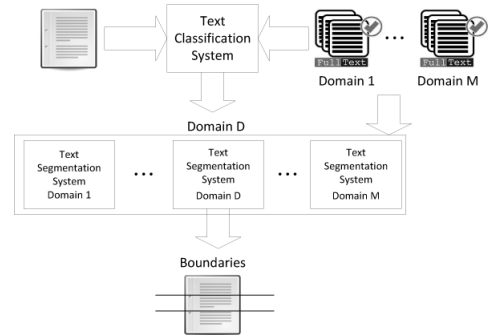


Figure 10. Connection of Text Segmentation with Text Clustering

The execution flow of text segmentation which is con-

nected with the text clustering is illustrated in Figure 11. Unlabeled texts are clustered into subgroups, each of which is a domain. Sample paragraph pairs which are labeled with boundary or continuance are generated from each domain. These sample paragraph pairs are provided for training the classifier in each text segmentation system. Each classifier is used for judging whether the topic change happens between paragraphs in each pair, or not.

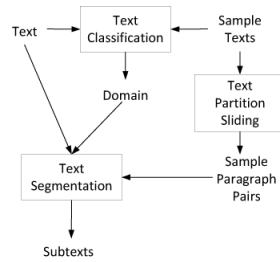


Figure 11. Execution Flow of Text Segmentation connected with Text Clustering

Let us make some remarks on the connection of the text segmentation with the text clustering. It is the process of segmenting a text group into subgroups based on their content-based similarities. The role of text clustering is to define the domains initially in the architecture of connecting the text segmentation with the text clustering. In each domain, sample paragraph pairs are generated, and its corresponding classifier is trained with them. In the future research, we consider using the text segmentation for clustering texts in the opposite direction.

REFERENCES

- [1] T. Jo, "Graph based KNN for Optimizing Index of News Articles", 53-62, Journal of Multimedia Information System, 3, 2016.
- [2] T. Jo, "Comparing Graph based K Nearest Neighbor with Traditional Version in Word Categorization in NewsPage.com, 12-18, International Journal of Advanced Social Sciences, 1, 2018.
- [3] T. Jo, "Specializing K Nearest Neighbor for Content based Segmentation of News Article by Graph Similarity Metric", 9-12, The Proceedings of International Conference on Green and Human Information Technology Part II, 2019.
- [4] T. Jo, "Graph based Version for Clustering Texts in Current Affair Domain", 171-176, The Proceedings of 15st International Conference on Data Science, 2019.